

TARDEC

---TECHNICAL REPORT---

No. **14351**

By: Wesley Bylsma



CREATION OF VIRTUAL REALITY MODELING LANGUAGE (VRML) DISPLACEMENT DATA FROM PAR DATA

Distribution: Approved for public release; distribution is unlimited.

U.S. Army Research, Development and Engineering Command
U.S. Army Tank-automotive and Armaments Research
Development and Engineering Center
Detroit Arsenal
6501 East 11 Mile Road
Warren, Michigan 48397-5000

Report Documentation Page			Form Approved OMB No. 0704-0188		
Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.					
1. REPORT DATE 01 NOV 2004		2. REPORT TYPE		3. DATES COVERED -	
4. TITLE AND SUBTITLE CREATION OF VIRTUAL REALITY MODELING LANGUAGE (VRML) DISPLACEMENT DATA FROM PAR DATA				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S) WESLEY BYLSMA				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) US ARMY TARDEC - NATIONAL AUTOMOTIVE CENTER,ATTN: AMSRD-TAR-N/MS157,6501 EAST 11 MILE RD,WARREN,MI,48397-5000				8. PERFORMING ORGANIZATION REPORT NUMBER 14351	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT A process for converting Par body position/orientation files to generic ".displacement" files using the AWK programming language is presented, with the intent of future use in scene assembly into a Virtual Reality Modeling Language (VRML) file.					
15. SUBJECT TERMS					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES 11	19a. NAME OF RESPONSIBLE PERSON
a. REPORT unclassified	b. ABSTRACT unclassified	c. THIS PAGE unclassified			

1.0 INTRODUCTION.....	1
2.0 PAR FORMAT.....	2
3.0 DISPLACEMENT FORMAT.....	3
4.0 VRML FORMAT.....	3
5.0 GAWK CONVERSION.....	5
5.1 STEP 1 INPUT	5
5.2 STEP 1 PARAMETERS.....	6
5.3 STEP 1 OUTPUT	6
5.4 STEP 1 CODE SECTIONS	6
5.5 STEP 2 INPUT	6
5.6 STEP 2 PARAMETERS.....	7
5.7 STEP 2 OUTPUT	7
5.8 STEP 2 CODE SECTIONS	7
6.0 SUMMARY/CONCLUSION	8
CONTACT.....	8
REFERENCES.....	8
DEFINITIONS, ACRONYMS, ABBREVIATIONS.....	8
APPENDIX A – PARHDR.AWK SCRIPT.....	9
APPENDIX B – PAR2DISPLACEMENT.AWK SCRIPT.....	10
FIGURE 1 – ESSENTIAL VRML VISUALIZATION COMPONENTS.....	1
TABLE 1 - DISPLACEMENT FILE VARIABLE FORMAT DEFINITION	3
FIGURE 2 – GAWK PROCESS.....	5
TABLE 2 - PARAMETER DEFINITIONS.....	6
TABLE 3 - PARAMETER DEFINITIONS.....	7

CREATION OF VIRTUAL REALITY MODELING LANGUAGE (VRML) DISPLACEMENT DATA FROM PAR DATA

Wesley Bylsma

U.S. Army Research, Development and Engineering Command (RDECOM)
U.S. Army Tank-automotive and Armaments Research, Development and Engineering Center (TARDEC)
National Automotive Center (NAC)
ATTN: AMSRD-TAR-N/MS157
6501 E 11 Mile Road
Warren, Michigan 48397-5000

1.0 INTRODUCTION

Visualization of complex information is one of the best ways to communicate its meaning. The focus of this effort is on the creation of the appearance portion a Virtual Reality Modeling Language (VRML) file that is used to visualize ground vehicle simulations. As Figure 1 depicts, there are five essential elements that should be included within the composite VRML file for meaningful visualization effects. Only the displacement element is discussed here.

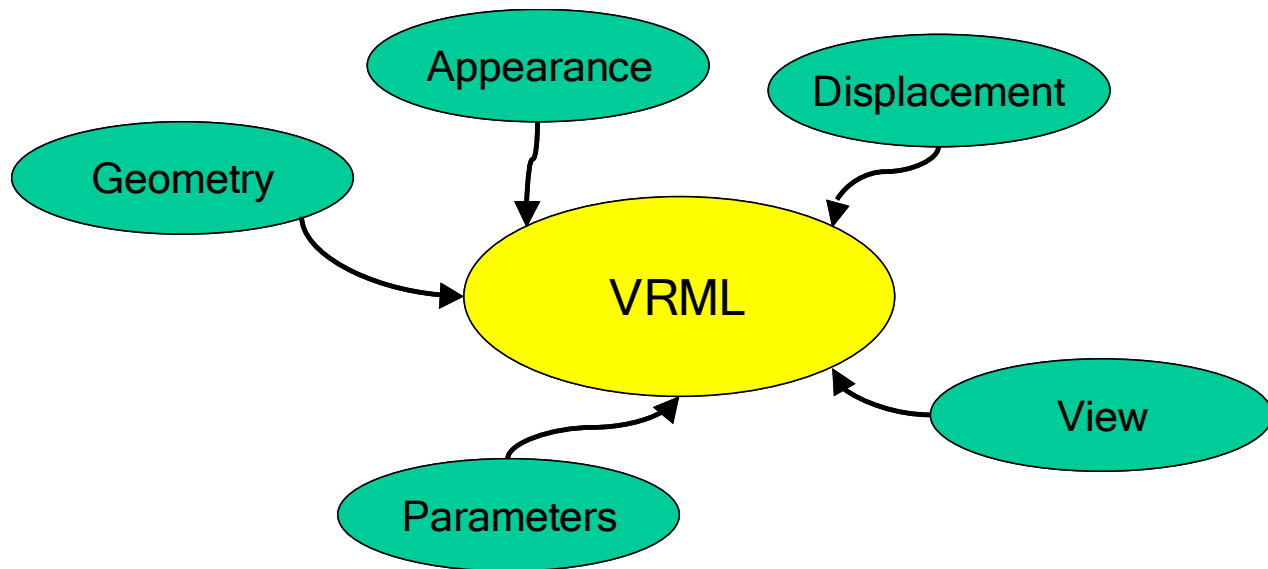


Figure 1 – Essential VRML Visualization Components

The displacement element is a generic file with the ".displacement" extension that contains position, orientation, and scale information for each geometry part. This report addresses the conversion process between the Par format to the generic ".displacement" format to be used with VRML. The conversion process is accomplished with the AWK Programming Language, named after its authors (Alfred V. Aho, Brian W. Kernighan, and Peter J. Weinberger at Bell Labs), which is designed to provide easy data manipulation and extraction of text files. In this case the Free Software Foundation's GNU version, GAWK, is used. More conversion processes may be developed in the future for inclusion of other displacement file formats. The focus of this discussion is restricted to the conversion of the Par format. Section 2.0 begins with a discussion of the Par format. Section 3.0 discusses the displacement (".displacement") format, section 4.0 discusses the VRML format that will be generated from the ".displacement" format, and section 5.0 outlines the GAWK conversion processes with subsections on specific topics.

2.0 PAR FORMAT

The Par or ".par" file format is output by DADS [1] for simulation results. It is a simple ASCII based format. It contains two sections. Section one defines the number of time steps, total number of bodies, and name of each body in the simulation. Section two defines the position and orientation of each body at each time step.

Below is an example file layout.

```
43 2226
TCH.BOD      TA1.BOD      TA2.BOD      TA3.BOD
TA4.BOD      TA5.BOD      TAILS.BOD    TAIRS.BOD
TA2LS.BOD    TA2RS.BOD    TA5LS.BOD    TA5RS.BOD
TAILW.BOD    TA1RW.BOD    TA2LW.BOD    TA2RW.BOD
TA3LW.BOD    TA3RW.BOD    TA4LW.BOD    TA4RW.BOD
TA5LW.BOD    TA5RW.BOD    TFIA.BOD     TRIA.BOD
TPA.BOD      TEQBFL.BOD   TEQBFR.BOD   TEQBRL.BOD
TEQBRR.BOD   TLOAD.BOD    HCH.BOD      HA1.BOD
HA2.BOD      HA3.BOD      HFRG.BOD     HA1LW.BOD
HA1RW.BOD    HA2LW.BOD    HA2RW.BOD    HA3LW.BOD
HA3RW.BOD    HDB.BOD     HLOAD.BOD

10.37358E+030.40454E-010.53814E+020.99999E+000.47363E-030.34105E-02-.29446E-04
20.46799E+030.33297E-010.23808E+020.99999E+000.20388E-040.39580E-02-.28594E-04
30.40800E+030.44312E-010.23790E+020.99999E+000.13526E-030.49973E-02-.27964E-04
40.27326E+030.14361E+000.24592E+020.10000E+010.24140E-040.29435E-02-.12050E-03
50.21428E+030.15696E+000.25025E+020.10000E+010.38804E-040.28499E-02-.30488E-04
60.15430E+03-.18947E+000.25028E+020.99999E+000.20740E-040.46665E-02-.30501E-04
70.46800E+030.39533E+020.23810E+020.99999E+000.48703E-040.40108E-02-.45811E-03
80.46798E+03-.39467E+020.23806E+020.99999E+000.48694E-040.39053E-02-.45803E-03
90.40799E+030.39544E+020.23800E+020.99999E+000.81364E-040.48952E-020.80261E-03
100.40801E+03-.39456E+020.23780E+020.99999E+000.81402E-040.50991E-020.80214E-03
110.15430E+030.39311E+020.25029E+020.99999E+000.10566E-040.46401E-020.18448E-03
120.15430E+03-.39689E+020.25027E+020.99999E+000.10564E-040.46929E-020.18456E-03
130.46800E+030.39533E+020.23540E+02-.70398E+00-.35860E-03-.71022E+000.28921E-03
140.46798E+03-.39467E+020.23536E+02-.70410E+00-.35849E-03-.71010E+000.28921E-03
150.40799E+030.39544E+020.23530E+02-.69794E+000.51488E-03-.71616E+00-.62108E-03
160.40801E+03-.39456E+020.23510E+02-.69757E+000.51473E-03-.71652E+00-.62059E-03
170.27326E+030.39644E+020.24324E+02-.73921E+00-.98782E-04-.67348E+000.73107E-04
180.27325E+03-.39356E+020.24320E+02-.73707E+00-.99013E-04-.67581E+000.72793E-04
190.21428E+030.39657E+020.24758E+02-.66130E+00-.48519E-04-.75012E+00-.88445E-05
200.21427E+03-.39343E+020.24752E+02-.59645E+00-.47612E-04-.80265E+00-.12859E-04
210.15430E+030.39311E+020.24760E+02-.61776E+000.13796E-03-.78637E+00-.12281E-03
220.15430E+03-.39689E+020.24757E+02-.65356E+000.13218E-03-.75687E+00-.12912E-03
230.43454E+030.18843E+020.47116E+020.10000E+010.47362E-030.30498E-02-.29616E-04
240.18841E+030.18061E+020.43994E+020.10000E+010.47358E-030.18229E-02-.30198E-04
250.50248E+030.96472E+010.38244E+020.10000E+010.47362E-030.29826E-02-.29648E-04
260.43793E+030.18040E+020.17002E+020.10000E+010.19660E-04-.11693E-03-.10481E-03
270.43793E+03-.17960E+020.16997E+020.10000E+010.19652E-04-.18584E-03-.10482E-03
280.18424E+030.17984E+020.18228E+020.10000E+010.55301E-040.19544E-040.28890E-02
290.18424E+03-.18016E+020.18226E+020.10000E+010.55270E-040.30416E-040.28890E-02
300.22548E+030.82580E-020.97525E+020.99999E+000.47363E-030.34105E-02-.29446E-04
31-.56790E+020.49190E-010.48439E+020.10000E+01-.22894E-040.23900E-030.18308E-03
320.39516E+020.79696E-010.25414E+020.10000E+01-.10104E-040.88866E-03-.75741E-04
33-.81968E+020.32293E-010.25296E+020.10000E+01-.88482E-050.23900E-030.18243E-03
34-.11999E+030.17978E-010.25280E+020.10000E+01-.90812E-050.23901E-030.18373E-03
350.42264E+020.84956E-010.37392E+020.10000E+01-.22956E-040.23899E-03-.79492E-04
360.42516E+020.40329E+020.21707E+020.49942E+00-.70687E-04-.86636E+00-.29063E-04
370.42504E+02-.40171E+020.21709E+020.52017E+00-.69966E-04-.85406E+00-.30759E-04
38-.80564E+020.40283E+020.21595E+020.40972E+000.16280E-03-.91221E+000.82798E-04
39-.80535E+02-.40217E+020.21596E+020.41002E+000.16277E-03-.91208E+000.82851E-04
40-.13101E+030.40264E+020.21585E+020.43043E+000.16194E-03-.90263E+000.87251E-04
41-.13098E+03-.40236E+020.21586E+020.43059E+000.16192E-03-.90255E+000.87282E-04
420.62764E+020.81697E-010.37382E+020.10000E+01-.22861E-040.14424E-02-.79519E-04
43-.74212E+020.44547E-010.86448E+020.10000E+01-.22894E-040.23900E-030.18308E-03
10.37915E+030.40961E-010.53813E+020.99999E+000.47399E-030.32759E-02-.24350E-04
20.47357E+030.36682E-010.23814E+020.99999E+000.13100E-040.37481E-02-.23578E-04
30.41358E+030.45053E-010.23798E+020.99999E+000.13185E-030.48672E-02-.22955E-04
...
```

The first line specifies the number of bodies and number of time steps (FORTRAN format of "I3,I6"). Each following line specifies the body name in multiples of 4 (FORTRAN format "4A20") until the number of bodies is reached. Beginning at the beginning of the next line, the body number, x, y, z, and euler parameters (e0,e1,e2,e3) are specified. This repeats for each time step (FORTRAN format ("I3,7E11.4")).

Note that conversion between euler parameters and quaternions is governed by

$$e0 = \cos\left(\frac{a}{2}\right)$$

$$(e1, e2, e3) = U \sin\left(\frac{a}{2}\right)$$

where U is a unit vector. So

$$a = 2 \cos^{-1}(e0) .$$

If $a < 0$ then it is subtracted from π to get a positive rotation angle.

3.0 DISPLACEMENT FORMAT

The displacement file is also a simple ASCII file like the Par file, only with a different format. Its format is laid out into one section encapsulated by a header and trailer line. Below is an example ".displacement" file:

```
# A-HA1.BOD
#NSTEPS: 2226
1.003706 0.002024 0.645516 0.000000 -0.000010 0.000889 -0.000076
1.145413 0.002060 0.647116 0.000000 -0.000004 0.001006 -0.000096
1.287602 0.002087 0.649605 0.000000 0.000001 0.000977 -0.000118
1.429893 0.002399 0.651866 0.000000 0.000015 0.000858 -0.000203
...
0.441985 0.001879 0.653466 0.000000 -0.000007 -0.000388 -0.000085
0.581939 0.001908 0.650418 0.000000 -0.000012 -0.000144 -0.000066
0.722046 0.001944 0.647421 0.000000 -0.000015 0.000231 -0.000058
0.862559 0.001985 0.645643 0.000000 -0.000013 0.000611 -0.000062
#END
```

The first line is a comment denoting the body name the data is for. The section begins with "#NSTEPS:" and ends with "#END". Following the semicolon character in "#NSTEPS:" the number of time steps should be given. Each line between the header and trailer contains displacement information for each time step. For each time step the position, orientation, and scale can be specified. A variable format exists because displacement information may not need each of these three components. Table 1 defines how each type is specified and what nodes are used for them (NF = number of fields in gawk).

Table 1 - Displacement File Variable Format Definition

NF	Type	Nodes
>=3	Position	PositionInterpolator
>=7	Position/Orientation	PositionInterpolator/OrientationInterpolator
>=10	Position/Orientation/Scale	PositionInterpolator/OrientationInterpolator/PositionInterpolator

The above example file contains position and orientation (NF >= 7) displacement data, but not scale data. Note that orientation data requires position data, and scale data requires position and orientation data.

4.0 VRML FORMAT

The displacement data used in the VRML file [2] is included using the PositionInterpolator (position and scale) and OrientationInterpolator nodes as defined in ISO 14772-1:1997. A TimeSensor node and ROUTE statement are also required. The creation of these nodes is done during final scene assembly with another program. A description is included here to help understand where the displacement data is included into the final VRML scene file.

The PositionInterpolator Node template is

DEF DPx PositionInterpolator { key ... keyvalue }.

This node defines the displacement position of the geometry. By including a name to the node definition, it can be referenced later in the ROUTE statement which is based off the TimeSensor node to give motion. This node is also used for scale information since it contains the same type of information---key values are sets of floating point sx,sy, and sz data.

The OrientationInterpolator Node template is

DEF DOx OrientationInterpolator { key ... keyvalue }

This node defines the orientation of the geometry. By including a name to the node definition, it can be referenced later in the ROUTE statement which is also based off the TimeSensor node to give rotational motion. This nodes key values are sets of four floating point numbers. From [2], the first three values specify a normalized rotation axis vector about which the rotation takes place. The fourth value specifies the amount of right-handed rotation about that axis in radians. The 3x3 matrix representation of a rotation (x y z a) is

$$\begin{bmatrix} tx2+c & txy+sz & txz-sy \\ txy-sz & ty2+c & tyz+sx \\ txz+sy & tyz-sx & tz2+c \end{bmatrix}$$

where $c = \cos(a)$, $s = \sin(a)$, and $t = 1-c$. An example section of a VRML (.wrl) file is included below:

```
DEF DP9 PositionInterpolator {
  key [
    0.000449,
    0.000898,
    0.001348,
    0.001797,
    ...
    0.998652,
    0.999102,
    0.999551,
    1.000000,
  ]
  keyValue [
    9.488932 0.001028 1.366876,
    9.630410 0.001040 1.366850,
    9.772396 0.001052 1.366749,
    9.914128 0.001065 1.366520,
    ...
    8.925052 0.000976 1.366444,
    9.065768 0.000988 1.366571,
    9.206738 0.000999 1.366698,
    9.347708 0.001013 1.366825,
  ]
}
DEF DO9 OrientationInterpolator {
  key [
    0.000449,
    0.000898,
    0.001348,
    0.001797,
    ...
    0.998652,
    0.999102,
    0.999551,
    1.000000,
  ]
  keyValue [
    0.008944 0.000474 0.003410 -0.000029,
    0.008944 0.000474 0.003276 -0.000024,
    0.008944 0.000471 0.003212 -0.000019,
    0.008944 0.000465 0.003240 -0.000016,
    ...
    0.008944 0.000429 0.003737 -0.000027,
    0.008944 0.000447 0.003771 -0.000032,
    0.008944 0.000460 0.003708 -0.000034,
    0.008944 0.000469 0.003573 -0.000033,
  ]
}
```

For each interpolator node there is an associated ROUTE statement that is based on the TimeSensor node. The ROUTE statement sets a routing path between the time change in the TimeSensor node to the interpolator node so that the displacement (position, orientation, or scale) can be looked up. This "looked up" value is then routed to a Transform node that encapsulates the geometry definition and performs translation, rotation, and scaling from the values routed to it. An example section of this part of the VRML (.wrl) file is included below:

```
DEF Tic TimeSensor { cycleInterval 73.33 loop TRUE }
...
```

```

Transform { translation 0 0 0 rotation 1 0 0 -1.57 children [
  DEF TG9 Transform { children [
    DEF S9_1 Shape {
      geometry USE C9_P1
      appearance USE A9_1
    }
    DEF S9_2 Shape {
      geometry USE C9_P2
      appearance USE A9_2
    }
    DEF S9_3 Shape {
      geometry USE C9_P3
      appearance USE A9_3
    }
    DEF S9_4 Shape {
      geometry USE C9_P4
      appearance USE A9_4
    }
  ] }
...
] }
...
ROUTE Tic.fraction_changed TO DP9.set_fraction
ROUTE DP9.value_changed TO TG9.set_translation
ROUTE Tic.fraction_changed TO DO9.set_fraction
ROUTE DO9.value_changed TO TG9.set_rotation
ROUTE Tic.fraction_changed TO DS9.set_fraction
ROUTE DS9.value_changed TO TG9.set_scale

```

In the above example, the "speed" of motion can be changed by modifying the cycleInterval value to more or less than the simulation time.

5.0 GAWK CONVERSION

The creation of the displacement portion of the VRML file is done with GAWK. This is a very useful scripting language that is available for UNIX and Windows operating systems from The Free Software Foundation ("www.gnu.org/software/gawk/gawk.html") or directly from Bell Labs ("cm.bell-labs.com/cm/cs/awkbook/"). Figure 2 outlines the AWK process.

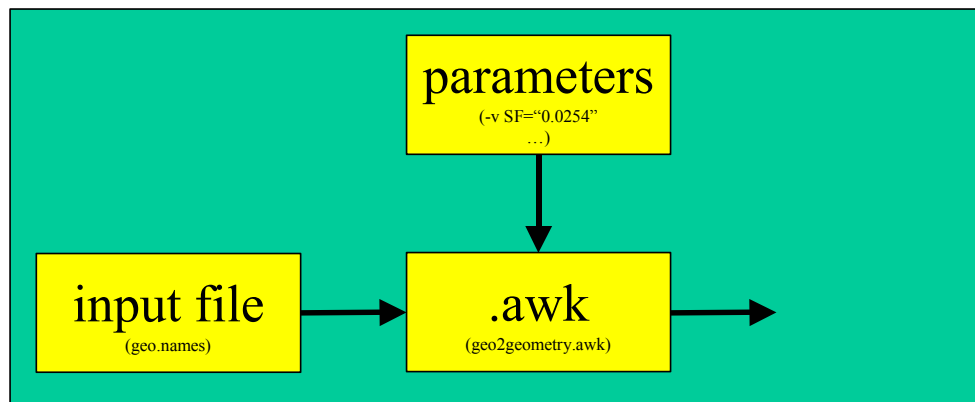


Figure 2 – GAWK process

Parameters are passed to the AWK script before it begins processing the input file. Output can be sent to the standard output or a specified file.

For conversion of Par files, the input process has two steps: 1) finding what bodies are available within a ".par" file and what their body numbers are, and 2) selection of which body information to extract and the actual extraction. Two separate scripts have been developed to accomplish each task.

5.1 STEP 1 INPUT

Step one is accomplished by the "parhdr.awk" script, written to extract the header information from the Par file. An example of the calling structure is given below

```
gawk -v FO="par.tmp" -v SRC="" -v SUF=".par" -f ../src/parhdr.awk par.names
```


The contents of the "par.names" file is a list of Par file names (excluding the extension). Its format contains two columns. The first column is the prefix string used for each body name within that Par file, and the second column is the Par file name. An example is included below:

```
A- plsttbtspreader-RMS1.50-10mph
B- plsttbtspreader-RMS1.50-10mph
```

A prefix string is needed to distinguish between body names that may be the same within different Par files (simulation runs) from the same model.

5.2 STEP 1 PARAMETERS

Table 2 defines and describes the parameters passed to the "parhdr.awk" script for processing the "par.names".

Table 2 - Parameter Definitions

VARIABLE	DESCRIPTION	EXAMPLE
FO	File Output Name	"par.tmp"
SRC	Source Directory.	""
SUF	Par File Suffix Extension.	".par"

It is assumed that all file names listed in the "par.names" input file have the same extension (in the example ".par").

5.3 STEP 1 OUTPUT

The output from the Par file header is written to the file specified in the FO parameter. Its format contains three columns within each section. Each section begins with "#PAR:" and ends with "#END". Following the semicolon character in the "#PAR:" header line the name of the Par file should be given. Each line between the header and trailer contains the line number (within that section), prefix and body name, and body number within the Par file. An example is included below:

```
#PAR: plsttbtspreader-RMS1.50-10mph.par
1  A-TCH.BOD 1
2  A-TA1.BOD 2
3  A-TA2.BOD 3
4  A-TA3.BOD 4
...
39 A-HA2RW.BOD 39
40 A-HA3LW.BOD 40
41 A-HA3RW.BOD 41
42 A-HDB.BOD 42
43 A-HLOAD.BOD 43
#END
#PAR: plsttbtspreader-RMS1.50-10mph.par
1  B-TCH.BOD 1
2  B-TA1.BOD 2
3  B-TA2.BOD 3
4  B-TA3.BOD 4
...
39 B-HA2RW.BOD 39
40 B-HA3LW.BOD 40
41 B-HA3RW.BOD 41
42 B-HDB.BOD 42
43 B-HLOAD.BOD 43
#END
```

5.4 STEP 1 CODE SECTIONS

The "parhdr.awk" script is included in Appendix A. Processing for each file name is continued as follows:

1. Read prefix and ".par" file name.
2. Read body names and number.
3. Write formatted output of ".par" body names and numbers.

Note that the output format is ("%3d %-30s %3d\n").

5.5 STEP 2 INPUT

Step two is accomplished by the "par2displacement.awk" script, written to extract the body position and orientation information from the Par file. An example of the calling structure is given below

```
gawk -v SF=0.0254-v DES="./testd/" -f ../src/par2displacement.awk par.tmp1
```

The contents of the "par.tmp1" file is an edited copy of the "par.tmp" Step 1 Output (see section 5.3). An example is included below:

```
#PAR: plsttbtspreader-RMS1.50-10mph.par
1  A-TCH.BOD 1
2  A-TA1.BOD 2
3  A-TA2.BOD 3
4  A-TA3.BOD 4
39 A-HA2RW.BOD 39
40 A-HA3LW.BOD 40
41 A-HA3RW.BOD 41
42 A-HDB.BOD 42
43 A-HLOAD.BOD 43
#END
#PAR: plsttbtspreader-RMS1.50-10mph.par
1  B-TCH.BOD 1
2  B-TA1.BOD 2
#END
```

Only columns two and three are important. Column one does not need to be renumbered.

5.6 STEP 2 PARAMETERS

Table 3 defines and describes the parameters passed to the "par2displacement.awk" script for processing the "par.tmp1" file.

Table 3 - Parameter Definitions

VARIABLE	DESCRIPTION	EXAMPLE
SF	Scale Factor. Conversion to meters.	0.0254
DES	Destination Directory.	"./testd/"

Note that the source directory will be included with the file name in the header line (see section 5.2 Step 1 Parameters).

5.7 STEP 2 OUTPUT

The converted file will have the same name as the body names (second column) in the "par.tmp1" file, but with a ".displacement" extension. This format is discussed in section 3.0.

While it is not considered directly part of the conversion process, output of a formatted version of the "par.tmp1" file is required for scene assembly as discussed in Section 1.0 and Figure 1. An example of this simple AWK program is given below:

```
gawk 'BEGIN {cnt=0;}{if (substr($0,1,1)!="#{cnt++;printf "%3d %-30s %-s\n", cnt,$2,$3;}}' par.tmp1 > D.names
```

The "D.names" file is used for visual inspection of the displacement index number to ensure proper alignment with other VRML properties and is used in final scene assembly. An example of the "D.names" file format is given below.

```
1  A-TCH.BOD 1
2  A-TA1.BOD 2
3  A-TA2.BOD 3
4  A-TA3.BOD 4
5  A-HA2RW.BOD 39
6  A-HA3LW.BOD 40
7  A-HA3RW.BOD 41
8  A-HDB.BOD 42
9  A-HLOAD.BOD 43
10 B-TCH.BOD 1
11 B-TA1.BOD 2
```

5.8 STEP 2 CODE SECTIONS

The "par2displacement.awk" script is included in Appendix B. Processing for each ".par" file is as follows:

1. Read in Par file name
2. Read in selected body information for Par file
3. Write header for each selected body displacement file
4. Write selected body position and orientation data after

5. Write trailer for each selected body displacement file

Note the format for the displacement file (for ".par" conversion) is ("%f %f %f %f %f %f %f\n"). Each position x,y, and z value is multiplied by the scale factor, SF, to convert into SI units. The rotation angle is normalized as discussed in section 2.0. The GAWK programming language does not have an inverse cosine function so use must be made of the inverse tangent function. The equivalent is given below

$$\cos^{-1}(x) = \tan^{-1}\left(\frac{\sqrt{1-x^2}}{x}\right), \quad x > 0$$

where the actual implementation in the code is

```
...  
a=2.0*atan2(sqrt(1.0-e0*e0),sqrt(e0*e0));  
if (a < 0) {a=3.14159-a;};  
...
```

6.0 SUMMARY/CONCLUSION

A simple script based conversion process between Par and a generic ".displacement" format was described for use in scene assembly of VRML files. It should be noted that the scene assembly portion mentioned in section 1.0 could be done with X3D [3]. Currently, however, many advanced utilities, such as Cortona Movie Maker [4] will only work with VRML and therefore is the focus at this time.

CONTACT

The author is an engineer with the U.S. Army Research, Development and Engineering Command (RDECOM), located at the U.S. Army Tank-automotive and Armaments Research, Development and Engineering Center (TARDEC). Interested parties can contact the author at the U.S. Army Tank-automotive and Armaments Research, Development and Engineering Center (TARDEC), ATTN: AMSRD-TAR-N/MS157, 6501 E 11 Mile Rd., Warren, Michigan 48397-5000, email: "bylsmaw@tacom.army.mil".

REFERENCES

- [1] Movie.BYU format, "lc.cray.com/doc/movie/".
- [2] The Virtual Reality Modeling Language (VRML), ISO/IEC 14772-1:1997 and 14772-2:2002, "www.web3d.org". (The Virtual Reality Modeling Language consists of two parts. Part 1 (ISO/IEC 14772-1) defines the base functionality and text encoding for VRML. Part 2 (ISO/IEC FDIS 14772-2) defines the base functionality and all bindings for the VRML External Authoring Interface).
- [3] X3D, ISO/IEC Draft 19776-1:200x, 19776-2:200x, 19777:200x, "www.web3d.org". (X3D encodings — ISO/IEC FDIS (Final Draft International Standard) 19776-1:200x (XML encoding) (.html) (.zip 220KB) 2004-09-26 Specifies the encoding of X3D files using the Extensible Markup Language (XML). X3D encodings — ISO/IEC FDIS (Final Draft International Standard) 19776-2:200x (Classic VRML encoding) (.html) (.zip 90KB) 2004-07-21 Specifies the encoding of the functionality and constructs defined in X3D using Classic VRML encoding. X3D language bindings — ISO/IEC FCD (Final Committee Draft) 19777:200x (.html) (.zip 143KB) 2003-05-15 Specifies the binding of the services in the X3D architecture to the ECMAScript programming language for use in X3D internal representation (Script nodes) and for external application access Specifies the binding of the services in the X3D architecture to the Java programming language for use in X3D internal representation (Script nodes) and for external application access).
- [4] Parallelgraphics, Inc. "www.parallelgraphics.com".
- [5] Dynamic Design and Analysis System (DADS) (now Virtual.Lab), LMS International, "www.lmsintl.com".

DEFINITIONS, ACRONYMS, ABBREVIATIONS

RDECOM – U.S. Army Research, Development and Engineering Center
TACOM - U.S. Army Tank-automotive and Armaments Command
TARDEC - TACOM Research, Development and Engineering Center
NAC - National Automotive Center
DADS - Dynamic Analysis and Design System
DIVA - DADS Interactive Visualizer

APPENDIX A – PARHDR.AWK SCRIPT

```
#
#parhdr.awk
#
# -v FO="par.tmp"
# -v SRC=""
# -v SUF=".par"
#
BEGIN {
    FS=" ";
}
{
    pre = $1;
    fi = SRC$2SUF;
    print fi,"->",FO;

    FIELDWIDTHS="3 6";
    err=getline < fi;
    if (err <=0) {print "---error reading",fi;exit;};
    nb = $1;
    nstps = $2;
# print nb,nstps;
#rem=print nb,(nb-(nb % 4))/4;
rem=nb % 4;
lms = (nb-rem)/4;
# if (rem !=0) lms++;
FIELDWIDTHS="20 20 20 20";
i=0;
print "#PAR:",fi > FO;
while (i<nb)
{
    err=getline<fi;
    if (err <=0) {print "---error reading",fi;exit;};
# print "NF",NF;
    for (j=1;j<=NF;j++)
    {
        i=i+1;
        if (j==1) bd[i] = $1;
        if (j==2) bd[i] = $2;
        if (j==3) bd[i] = $3;
        if (j==4) bd[i] = $4;
        printf "%3d %-30s %3d\n", i,pre""bd[i],i >> FO;
        if (i>=nb) break;
    }
}
print "#END" > FO;
close(fi);
FS = " "; # reset delimiter
}
```

APPENDIX B – PAR2DISPLACEMENT.AWK SCRIPT

```
#
#par2displacement.awk
#
# -v SF=0.254
# -v DES="./testd/"
#
BEGIN {
  cPAR="#PAR:";
  cEND="#END";
  FS = " ";
  while (err=getline > 0)
  {
    print "line:",$0;
    FS = " ";
    if (substr($0,1,length(cPAR))==cPAR)
    {
      fi = $2;
      print "Processing",fi;
      cnt=0;
      delete num;
      while (err=getline > 0 && substr($0,1,length(cEND))!=cEND)
      {
        cnt++;
        num[$3+0] = $2;
      }
      print "cnt=",cnt;
      FIELDWIDTHS="3 6";
      err = getline < fi ;
      if (err <= 0) { print "---error reading",fi;exit;};
      nb = $1;
      nsteps = $2;
      print nb,nsteps;
      rem=nb % 4;
      lns = (nb-rem)/4;
      FIELDWIDTHS="20 20 20 20";
      i=0;
      while (i<nb)
      {
        err = getline < fi;
        if (err <= 0) { print "---error reading",fi;exit;};
        for (j=1;j<=NF;j++)
        {
          i=i+1;
          if (i>=nb) break;
        }
      }
      for (i in num)
      {
        print "#",num[i] > DES"num[i].displacement";
        print "#NSTEPS:",nsteps >> DES"num[i].displacement";
        print "#NSTEPS:",nsteps, DES"num[i].displacement";
      }
      FIELDWIDTHS="3 11 11 11 11 11 11 11";
      while (err=getline < fi > 0)
      {
        # e0=cos(a/2);, e(e1,e2,e3)=U sin(a/2); so a=2*acos(e0) */
        # acos = atan(sqrt(1-x^2)/x ) for x > 0;
        tmp = $1+0;
        x=$2+0.0;
        y=$3+0.0;
        z=$4+0.0;
        e0=$5+0.0;
        e0tmp=2.0*atan2(sqrt(1.0-e0*e0),sqrt(e0*e0));
        if (e0 <0) {e0tmp=3.14159-e0tmp;};
        e1=$6+0.0;
        e2=$7+0.0;
        e3=$8+0.0;
        if (tmp in num)
        {
          printf "%f %f %f %f %f %f %f\n",SF*x,SF*y,SF*z,e1,e2,e3,e0tmp >> DES"num[tmp].displacement";
        }
      }
    }
  }
  for (i in num)
```

```
    {
      print "#END" >> DES""num[i]".displacement";
    }
    FS = " "; # reset
  } # if "#PAR:"
  close(fi);
} # end while
}
```